Active Learning of Recursive Functions by Ultrametric Algorithms⋆

Rūsiņš Freivalds[1] and Thomas Zeugmann[2]

[1] Institute of Mathematics and Computer Science, University of Latvia
Raiņa bulvāris 29, Riga, LV-1459, Latvia
Rusins.Freivalds@mii.lu.lv
[2] Division of Computer Science, Hokkaido University
N-14, W-9, Sapporo 060-0814, Japan
thomas@ist.hokudai.ac.jp

Abstract. We study active learning of classes of recursive functions by asking value queries about the target function f, where f is from the target class. That is, the query is a natural number x, and the answer to the query is $f(x)$. The complexity measure in this paper is the worst-case number of queries asked. We prove that for some classes of recursive functions *ultrametric active learning* algorithms can achieve the learning goal by asking *significantly fewer* queries than deterministic, probabilistic, and even nondeterministic active learning algorithms. This is the first ever example of a problem where ultrametric algorithms have advantages over nondeterministic algorithms.

1 Introduction

Inductive inference has been studied intensively. Gold [12] defined *learning in the limit*. The learner is a deterministic algorithm called *inductive inference machine* (abbr. IIM), and the objects to be learned are recursive functions. The information source are growing initial segments $(x_0, f(x_0)), \ldots, (x_n, f(x_n))$ of ordered pairs of the graph of the target function f. It is assumed that every pair $(x, f(x))$ appears eventually. As a hypothesis space one can choose any *Gödel numbering* $\varphi_0, \varphi_1, \varphi_2, \ldots$ of the set of all partial recursive functions over the natural numbers $\mathbb{N} = \{0, 1, 2, \ldots\}$ (cf. [27]). If an $i \in \mathbb{N}$ is such that $\varphi_i = f$ then we call i a *φ-program* of f. An IIM, on input an initial segment $(x_0, f(x_0)), \ldots, (x_n, f(x_n))$, has to output a natural number i_n which is interpreted as φ-program. An IIM *learns* f if the sequence $(i_n)_{n\in\mathbb{N}}$ of all computed φ-programs converges to a program i such that $\varphi_i = f$.

Every IIM M learns some set of recursive functions which is denoted by $EX(M)$. The family of all such sets, over the universe of effective algorithms viewed as IIMs, serves as a characterization of the learning power inherent in

⋆ The research was supported by Grant No. 09.1570 from the Latvian Council of Science and the Invitation Fellowship for Research in Japan S12052 by Japan Society for the Promotion of Science

the Gold model. This family is denoted by *EX* (short for explanatory) and it is defined by $EX = \{\mathcal{U} \mid \exists M(\mathcal{U} \subseteq EX(M)\}$. Many studies of inductive inference set-theoretically compare the family *EX* with families that arise from considering other models (cf., e.g., [30]). One such model is *finite learning*, where the IIM either requests a new input and outputs nothing, or it outputs a program i, and stops. Again we require that program i is correct for f, i.e., $\varphi_i = f$.

The models described so far are models of *passive* learning, since the IIM has no influence on the order in which examples are presented. In contrast, the learning model considered in the present paper is an *active* one. This model goes back to Angluin [3] and is called *query learning*. In the query learning model the learner has access to a teacher that truthfully answers queries of a prespecified type. In this paper we only consider *value queries*. That is, the query is a natural number x, and the answer to the query is $f(x)$. A query learner is an algorithmic device that, depending on the answers already received, either computes a new value query or it returns a hypothesis i and stops. As above, the hypothesis is interpreted with respect to a fixed Gödel numbering φ and it is required that the hypothesis returned satisfies $\varphi_i = f$. So active learning is *finite learning*.

As in the Gold [12] model, we are interested in active learners that can infer whole classes of recursive functions. The complexity measure is then the *worst-case number of queries asked* to identify all the functions from the target class $\mathcal{U}$. We refer to any query learner as *query inference machine* (abbr. QIM).

Automata theory and complexity theory have considered several natural generalizations of deterministic algorithms, namely, nondeterministic and probabilistic algorithms. In many cases these generalized algorithms allow for computations having a complexity that is strictly less than their deterministic counterpart. Such generalized algorithms attracted considerable attention in learning theory, too. Many papers studied learnability by nondeterministic algorithms [1, 5, 11, 29] and probabilistic algorithms [14, 17, 21, 22, 25, 26].

Definition 1. *We say that a nondeterministic QIM* learns a function f *if*

(1) *there is at least one computation path such that the QIM produces a correct result on f, i.e., a program j such that $\varphi_j = f$;*
(2) *at no computation path the QIM produces an incorrect result on f.*

Definition 2. *We say that a probabilistic QIM* learns a function f with a probability p *if*

(1) *the sum of all probabilities of all leaves which produce a correct result on f, i.e., a number j such that $\varphi_j = f$, is no less than p;*
(2) *at no computation path the QIM produces an incorrect result on f.*

Recently, Freivalds [7] introduced a new type of indeterministic algorithms called *ultrametric* algorithms. An extensive research on ultrametric algorithms of various kinds is performed by him and his co-authors (cf. [4, 15]). So, ultrametric algorithms are a very new concept and their potential still has to be explored. This is the first paper showing a problem where ultrametric algorithms *have advantages* over nondeterministic algorithms. Ultrametric algorithms are very similar to probabilistic algorithms but while probabilistic algorithms use *real* numbers r with $0 \leq r \leq 1$ as parameters, ultrametric algorithms use *p-adic* numbers as parameters. The usage of p-adic numbers as *amplitudes* and the ability to perform *measurements* to transform amplitudes into real numbers are inspired by quantum computations and allow for algorithms not possible in classical computations. Slightly simplifying the description of the definitions, one can say that ultrametric algorithms are the same as probabilistic algorithms, only the *interpretation* of the probabilities is *different*.

The choice of p-adic numbers instead of real numbers is not quite arbitrary. Ostrowski [24] proved that any non-trivial absolute value on the rational numbers $\mathbb{Q}$ is equivalent to either the usual real absolute value or a p-adic absolute value. This result shows that using p-adic numbers was not merely one of many possibilities to generalize the definition of deterministic algorithms but rather the only remaining possibility not yet explored.

The notion of p-adic numbers is widely used in science. String theory [28], chemistry [19] and molecular biology [6, 16] have introduced p-adic numbers to describe measures of indeterminism. Indeed, research on indeterminism in nature has a long history. Pascal and Fermat believed that every event of indeterminism can be described by a real number between 0 and 1 called *probability*. Quantum physics introduced a description in terms of complex numbers called *amplitude of probabilities* and later in terms of probabilistic combinations of amplitudes most conveniently described by *density matrices*. Using p-adic numbers to describe indeterminism allows to explore some aspects of indeterminism but, of course, does not exhaust all the aspects of it.

There are many distinct p-adic absolute values corresponding to the many prime numbers p. These absolute values are traditionally called *ultrametric*. Absolute values are needed to consider *distances* among objects. We are used to rational and irrational numbers as measures for distances, and there is a psychological difficulty to imagine that something else can be used instead of rational and irrational numbers, respectively. However, there is an important feature that distinguishes p-adic numbers from real numbers. Real numbers (both rational and irrational) are linearly ordered, while p-adic numbers *cannot* be linearly ordered. This is why *valuations* and *norms* of p-adic numbers are considered.

The situation is similar in Quantum Computation (see [23]). Quantum amplitudes are complex numbers which also cannot be linearly ordered. The counterpart of valuation for quantum algorithms is *measurement* translating a complex number $a + bi$ into a real number $a^2 + b^2$. Norms of p-adic numbers are rational numbers. We continue with a short description of p-adic numbers.

2 p-adic Numbers and Ultrametric Algorithms

Let p be an arbitrary prime number. A number $a \in \mathbb{N}$ with $0 \leq a \leq p - 1$ is called a *p-adic digit*. A *p-adic integer* is by definition a sequence $(a_i)_{i \in \mathbb{N}}$ of p-adic digits. We write this conventionally as $\cdots a_i \cdots a_2 a_1 a_0$, i.e., the a_i are written from left to right.

If n is a natural number, and $n = \overline{a_{k-1}a_{k-2}\cdots a_1 a_0}$ is its p-adic representation, i.e., $n = \sum_{i=0}^{k-1} a_i p^i$, where each a_i is a p-adic digit, then we identify n with the p-adic integer (a_i), where $a_i = 0$ for all $i \geq k$. This means that the natural numbers can be identified with the p-adic integers $(a_i)_{i\in\mathbb{N}}$ for which all but finitely many digits are 0. In particular, the number 0 is the p-adic integer all of whose digits are 0, and 1 is the p-adic integer all of whose digits are 0 except the right-most digit a_0 which is 1.

To obtain p-adic representations of all rational numbers, $\frac{1}{p}$ is represented as $\cdots 00.1$, the number $\frac{1}{p^2}$ as $\cdots 00.01$, and so on. For any p-adic number it is allowed to have infinitely many (!) digits to the left of the "p-adic" point but only a finite number of digits to the right of it.

However, p-adic numbers are not merely a generalization of rational numbers. They are related to the notion of *absolute value* of numbers. If X is a nonempty set, a distance, or metric, on X is a function d from $X \times X$ to the nonnegative real numbers such that for all $(x, y) \in X \times X$ the following conditions are satisfied.

(1) $d(x, y) \geq 0$, and $d(x, y) = 0$ if and only if $x = y$,
(2) $d(x, y) = d(y, x)$,
(3) $d(x, y) \leq d(x, z) + d(z, y)$ for all $z \in X$.

A set X together with a metric d is called a *metric space*. The same set X can give rise to many different metric spaces. If X is a linear space over the real numbers then the *norm* of an element $x \in X$ is its distance from 0, i.e., for all $x, y \in X$ and α any real number we have:

(1) $\|x\| \geq 0$, and $\|x\| = 0$ if and only if $x = 0$,
(2) $\|\alpha \cdot y\| = |\alpha| \cdot \|y\|$,
(3) $\|x + y\| \leq \|x\| + \|y\|$.

Note that every norm induces a metric d, i.e., $d(x, y) = \|x - y\|$. A well-known example is the metric over $\mathbb{Q}$ induced by the ordinary absolute value. However, there are other norms as well. A norm is called *ultrametric* if Requirement (3) can be replaced by the stronger statement: $\|x+y\| \leq \max\{\|x\|, \|y\|\}$. Otherwise, the norm is called *Archimedean*.

Definition 3. *Let $p \in \{2, 3, 5, 7, 11, 13, \ldots\}$ be any prime number. For any nonzero integer a, let the p-adic* ordinal (*or* valuation) *of a, denoted $ord_p\, a$, be the highest power of p which divides a, i.e., the greatest number $m \in \mathbb{N}$ such that $a \equiv 0 \pmod{p^m}$. For any rational number $x = a/b$ we define $ord_p\, x =_{df} ord_p\, a - ord_p\, b$. Additionally, $ord_p\, x =_{df} \infty$ if and only if $x = 0$.*

For example, let $x = 63/550 = 2^{-1} \cdot 3^2 \cdot 5^{-2} \cdot 7^1 \cdot 11^{-1}$. Thus, we have

$$\begin{array}{ll}
\mathrm{ord}_2\, x = -1 & \mathrm{ord}_7\, x = +1 \\
\mathrm{ord}_3\, x = +2 & \mathrm{ord}_{11}\, x = -1 \\
\mathrm{ord}_5\, x = -2 & \mathrm{ord}_p\, x = 0 \quad \text{for every prime } p \notin \{2, 3, 5, 7, 11\}\ .
\end{array}$$

Definition 4. *Let $p \in \{2, 3, 5, 7, 11, 13, \ldots\}$ be any prime number. For any rational number x, we define its* p-norm *as $p^{-ord_p\, x}$, and we set $\|0\|_p =_{df} 0$.*

For example, with $x = 63/550 = 2^{-1} \cdot 3^2 \cdot 5^{-2} \cdot 7^1 \cdot 11^{-1}$ we obtain:

$$\begin{array}{ll} \|x\|_2 = 2 & \|x\|_7 = 1/7 \\ \|x\|_3 = 1/9 & \|x\|_{11} = 11 \\ \|x\|_5 = 25 & \|x\|_p = 1 \quad \text{for every prime } p \notin \{2,3,5,7,11\} \ . \end{array}$$

Rational numbers are p-adic integers for all prime numbers p. Since the definitions given above are all we need, we finish our exposition of p-adic numbers here. For a more detailed description of p-adic numbers we refer to [13, 18].

We continue with *ultrametric algorithms*. In the following, p always denotes a prime number. Ultrametric algorithms are described by finite directed acyclic graphs (abbr. DAG), where exactly one node is marked as root. As usual, the root does not have any incoming edge. Furthermore, every node having outdegree zero is said to be a *leaf*. The leaves are the output nodes of the DAG.

Let v be a node in such a graph. Then each outgoing edge is labeled by a p-adic number which we call *amplitude*. We require that the sum of all amplitudes that correspond to v is 1. In order to determine the *total amplitude* along a computation path, we need the following definition.

Definition 5. *The total amplitude of the root is defined to be* 1*. Furthermore, let v be a node at depth d in the DAG, let α be its total amplitude, and let $\beta_1, \beta_2, \cdots, \beta_k$ be the amplitudes corresponding to the outgoing edges $e_1, \ldots, e_k$ of v. Let $v_1, \ldots, v_k$ be the nodes where the edges $e_1, \ldots, e_k$ point to. Then the total amplitude of v_ℓ, $\ell \in \{1, \ldots, k\}$, is defined as follows.*

(1) *If the indegree of v_ℓ is one, then its total amplitude is $\alpha\beta_\ell$.*
(2) *If the indegree of v_ℓ is bigger than one, i.e., if two or more computation paths are joined, say m paths, then let $\alpha, \gamma_2, \ldots, \gamma_m$ be the corresponding total amplitudes of the predecessors of v_ℓ and let $\beta_\ell, \delta_2, \ldots, \delta_m$ be the amplitudes of the incoming edges The total amplitude of the node v_ℓ is then defined to be $\alpha\beta_\ell + \gamma_2\delta_2 + \cdots + \delta_m\gamma_m$.*

Note that the total amplitude is a p-adic integer.

We refer the reader to the proof of Theorem 7 for an example.

It remains to define what is meant by saying that a p-ultrametric algorithm produces a result with a certain probability. This is specified by performing a so-called *measurement* at the leaves of the corresponding DAG. Here by measurement we mean that we transform the total amplitude β of each leaf to $\|\beta\|_p$. We refer to $\|\beta\|_p$ as the *p-probability* of the corresponding computation path.

Definition 6. *We say that a p-ultrametric algorithm* produces a result m with a p-probability q *if the sum of the p-probabilities of all leaves which correctly produce the result m is no less than q.*

Definition 7. *We say that a p-ultrametric QIM* learns a function f with a p-probability q *if*

(1) *the sum of the* p-probabilities *of all leaves which produce a correct result on f, i.e., a number j such that $\varphi_j = f$, is no less than q;*
(2) *at no computation path the QIM produces an incorrect result on f.*

3 Results

As explained in the Introduction we are interested in the number of queries a QIM has to ask in the worst-case in order to infer all recursive functions from a prespecified class $\mathcal{U}$. The hypothesis space will always be a Gödel numbering φ (cf. [27]). This is no restriction of generality since all natural programming languages provide Gödel numberings of recursive functions.

The complexity of learning recursive functions has been an important topic for several decades [2, 8, 10, 30]. In this paper we compare the query complexity of deterministic, nondeterministic, probabilistic, and ultrametric QIMs.

Our results are somewhat unexpected. Usually, for various classes of problems, nondeterministic algorithms provide the smallest complexity, deterministic algorithms provide the largest complexity and probabilistic algorithms provide some medium complexity. In [4, 7, 15] ultrametric algorithms also gave medium complexity sometimes better and sometimes worse than probabilistic algorithms. Our results in this paper show that, for learning recursive functions from value queries, there are classes $\mathcal{U}$ of recursive functions such that ultrametric QIMs have a much smaller complexity than even nondeterministic QIMs.

To show these results we use a combinatorial structure called the *Fano plane*. It is one of *finite geometries* (see [20]). The Fano plane consists of seven *points* $0, 1, 2, 3, 4, 5, 6$ and seven *lines* $(0, 1, 3)$, $(1, 2, 4)$, $(2, 3, 5)$, $(3, 4, 6)$, $(4, 5, 0)$, $(5, 6, 1)$, $(6, 0, 2)$. For any two points i, j with $i \neq j$, in this geometry there is exactly one line that contains these points (cf. Figure 1). For any two different lines in this geometry there is exactly one point contained in these two lines. In our construction the points $0, 1, 2, 3, 4, 5, 6$ are interpreted as colored in two colors RED and BLUE, respectively.

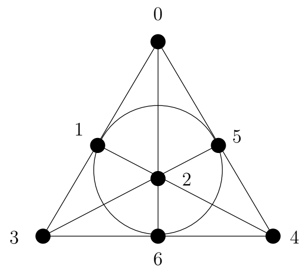

Fig. 1. The Fano Plane

Lemma 1 ([20]). *For an arbitrary coloring of the Fano plane there is at least one line the 3 points of which are colored by the same color.*

Lemma 2 ([20]). *For any coloring of the Fano plane there cannot exist two lines colored in opposite colors.*

Proof. Any two lines intersect at some point. □

To simplify notation, in the following we use $\mathcal{P}$ and $\mathcal{R}$ to denote the set of all partial recursive functions and of all recursive functions of one variable over $\mathbb{N}$, respectively. Let φ be a Gödel numbering of $\mathcal{P}$. We consider the following class $\mathcal{U}_7$ of recursive functions. Each function $f \in \mathcal{U}_7$ is such that $f \in \mathcal{R}$ and:

(1) every $f(x)$ where $0 \leq x \leq 6$ equals either 2^s or 3^t, where $s, t \in \mathbb{N}$, $s, t \geq 1$,
(2) if $0 \leq x_1 < x_2 \leq 6$, $f(x_1) = 2^s$ and $f(x_2) = 2^t$, then $f(x_1) = f(x_2)$,

(3) if $0 \leq x_1 < x_2 \leq 6$, $f(x_1) = 3^s$ and $f(x_2) = 3^t$, then $f(x_1) = f(x_2)$,
(4) there is a line (i, j, k) in the Fano plane such that $f(i) = f(j) = f(k) = 2^s$ and $\varphi_s = f$ or there exists a line (i, j, k) in the Fano plane such that $f(i) = f(j) = f(k) = 3^t$ and $\varphi_t = f$.

Comment. In our construction of the class $\mathcal{U}_7$ the points $0, 1, 2, 3, 4, 5, 6$ can be interpreted as colored in two colors. Some points $f(i)$ are such that $f(i) = 2^s$ (these points are described below as RED) while some other points j are such that $f(j) = 3^t$ (these points are described below as BLUE). The properties of the Fano plane ensure that for every such coloring in two colors there exists a line such that the three points on this line are colored in the same color, and there cannot exist two lines colored in opposite colors.

Definition 8. *A* partial coloring *C of a Fano plane is an assignment of colors RED, BLUE, NONE to the points of the Fano plane.*

A partial coloring C_2 is an extension *of a partial coloring C_1 if every point colored RED or BLUE in C_1 is colored in the same color in C_2.*

A partial coloring *C of a Fano plane is called* complete *if every point is colored RED or BLUE.*

Lemma 3. *Given any partial coloring C of the points in the Fano plane assigning colors RED and BLUE to some but not all points such that no line contains three points in the same color, there exists*

(1) *a complete extension of the given coloring C such that it contains a line with three RED points, and*
(2) *a complete extension of the given coloring C such that it contains a line with three BLUE points.*

Proof. Color all the not colored points RED for the first function, and BLUE for the second function. □

Lemma 4. *Given any partial coloring C of points in the Fano plane assigning colors RED and BLUE to some but not all points such that no line contains 3 points in the same color, there exist numbers $k_*, \ell_* \in \mathbb{N}$ and*

(1) *a function $f_{RED} \in \mathcal{U}_7$ defined as $f(x) = 2^{\ell_*}$ for all x colored RED in C such that f_{RED} contains a line with three RED points, and all the points $0, 1, 2, 3, 4, 5, 6$ are colored RED or BLUE,*
(2) *a function $f_{BLUE} \in \mathcal{U}_7$ defined as $f(x) = 3^{k_*}$ for all x colored BLUE in C such that f_{BLUE} contains a line with three BLUE points, and all the points $0, 1, 2, 3, 4, 5, 6$ are colored RED or BLUE.*

Proof. The assertions of the lemma can be shown by using the fixed point theorem [27] and by using Lemma 3. □

Theorem 1. *There is a deterministic QIM M that learns the class $\mathcal{U}_7$ with 7 queries.*

Proof. The desired QIM M queries the points 0, 1, ..., 6. After having received $f(0)$, $f(1)$, $f(2)$, ..., $f(6)$, it checks at which line all points have the same color, and outputs the φ-program corresponding to this line. Note that by Lemmata 1 and 2 there is precisely one such line. By the definition of the class $\mathcal{U}_7$ one can directly output a correct φ-program for the target function f. $\square$

Theorem 2. *There exists no deterministic QIM learning $\mathcal{U}_7$ with 6 queries.*

Proof. The proof is by contradiction. Using Smullyan's double fixed point theorem [27] one can construct two functions f and $\tilde{f}$ such that both are in $\mathcal{U}_7$ but at least one of them is not correctly learned by the QIM M. $\square$

Theorem 3. *There is a nondeterministic QIM M learning $\mathcal{U}_7$ with 3 queries.*

Proof. The QIM M starts with a nondeterministic branching of the computation into 7 possibilities corresponding to the 7 lines in the Fano plane. In each case, all 3 points i, j, k are queried. If $f(i), f(j), f(k)$ are not of the same color then the computation path is aborted. If they are of the same color, e.g., $f(i) = 2^{s_i}, f(j) = 2^{s_j}, f(k) = 2^{s_k}$ then the definition of the class $\mathcal{U}_7$ ensures that $s_i = s_j = s_k$ and the QIM M outputs s_i which is a correct program computing the function f. $\square$

Theorem 4. *There is no nondeterministic QIM learning $\mathcal{U}_7$ with 2 queries.*

Proof. By Lemma 4, there are two distinct functions in the class $\mathcal{U}_7$ with the same values queried by the nondeterministic algorithm. The output is not correct for at least one of them. $\square$

Theorem 5. *There is a probabilistic QIM M learning $\mathcal{U}_7$ with probability $\frac{1}{7}$ with 3 queries.*

Proof. The algorithm starts with branching its computation into 7 possibilities corresponding to the 7 lines in Fano plane. Each branch is reached with probability 1/7. In each branch, all 3 points i, j, k are queried. If $f(i), f(j), f(k)$ are not of the same color then the computation path is aborted. If they are of the same color, e.g., $f(i) = f(j) = f(k) = 2^s$, then s is output. By definition of the class $\mathcal{U}_7$ the result is a correct program computing the function f. $\square$

Theorem 6. *There is a probabilistic QIM M learning $\mathcal{U}_7$ with probability $\frac{4}{7}$ with 6 queries.*

Theorem 7. *For every prime number p, there is a p-ultrametric QIM M learning $\mathcal{U}_7$ with p-probability* 1 *with 2 queries.*

Proof. The desired QIM M branches its computation path into 7 branches at the root, where each branch corresponds to exactly one line of the Fano plane. We assign to each edge the amplitude 1/7. At the second level, each of these branches is branched into 3 subbranches each of which is assigned the amplitude 1/3. So far we have at level three 21 nodes denoted by $v_1, \dots, v_{21}$ (cf. Figure 2). For each of these nodes we formulate two queries. Let v be such that its father

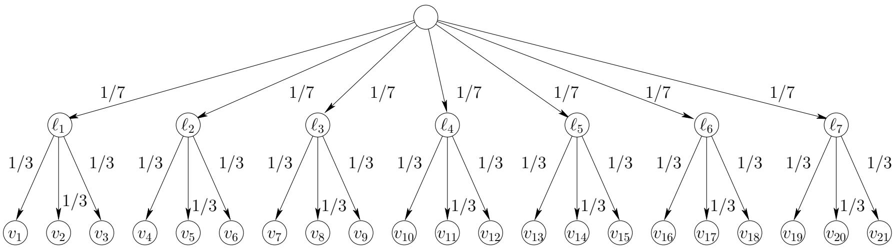

Fig. 2. The first three levels of the DAG representing the computation of the QIM M

node corresponds to the line containing the point i, j, k of the Fano plane, where we order these points such that $i < j < k$. If v is the leftmost node then we query (i, j), if v is the middle node then we query (j, k) and if v is the rightmost node then we query (i, k). Every triple of nodes having the same father share a register, say r_{ijk}. Initially, the register contains the value $\uparrow$ which stands for "no output." The node activated when reached in the computation path sends the following value to r_{ijk}. After having received the answer to its queries, e.g., $f(i) = 2^s$ and $f(j) = 3^t$ then it writes 0 in r_{ijk}, and if the values coincide, e.g., $f(i) = 3^t$ and $f(j) = 3^t$, then it writes t in r_{ijk}.

Looking at any triple of nodes having a common father at the third level, we note that the following 8 cases may occur as answer. We use again the corresponding colors, where R and B are shortcuts for RED and BLUE, respectively.

(i, j)	(j, k)	(i, k)	(i, j)	(j, k)	(i, k)
(B, B)	(B, R)	(B, R)	(R, R)	(R, B)	(R, B)
(B, B)	(B, B)	(B, B)	(R, R)	(R, R)	(R, R)
(B, R)	(R, R)	(B, R)	(R, B)	(B, B)	(R, B)
(B, R)	(R, B)	(B, B)	(R, B)	(B, R)	(R, R)

Thus, we need for each node at the third level 8 outgoing edges as the table above shows. If the edge corresponds to a pair (R, R) or (B, B) then we assign the amplitude $1/2$ and otherwise the amplitude $-1/4$. Note that sum of these amplitudes is again 1.

Finally, we join each triple as shown in table above into one node, e.g., the edges corresponding to (B, B), (B, R), and (B, R) are joined. If the total amplitude of such a node at the third level is different from zero, then the node produces as output the value stored in register r_{ijk}. Figure 3 shows the part of the DAG for the queries performed for the first line of the Fano plane, i.e., for the line $(0, 1, 3)$. So this part starts at the nodes v_1, v_2 and v_3 shown in Figure 2. For the sake of readability, we show the queries asked at each node, i.e., $(0, 1)$ at node v_1, $(1, 3)$ at node v_2, and $(0, 3)$ at node v_3. A dashed (blue) edge denotes the case that both answers to the queries asked at the corresponding vertex returned a value of f indicating that the related nodes of the first line of the Fano plane are blue. This result is then propagated along the dashed (blue) edges. Analogously, a dotted (red) edge indicates that both answers corresponded to a red node of the first line of the Fano plane. If the answers returned function values indicating that the colors of the queried nodes of the first line of the Fano plane have different colors then the edge is drawn in black. Dashed (blue) and dotted (red) edges have the amplitude $1/2$ and the black edges have the the amplitude $-1/4$.

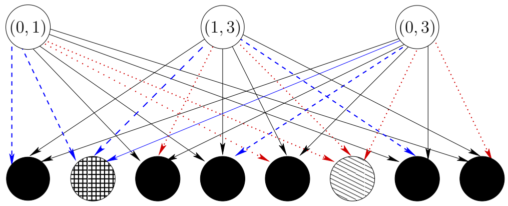

Fig. 3. The part of the DAG representing the computation of the QIM M for the line $(0, 1, 3)$ starting at the nodes of the third level

It remains to show that the QIM M has the desired properties. By construction, at every computation path exactly two queries are asked.

Next, by Definition 5 it is obvious that the total amplitude of each node at the second level is $1/21$. Next, we consider any node at the third level. If a triple (B, B), (B, B), and (B, B) is joined then the total amplitude is

$$\frac{1}{21} \cdot \frac{1}{2} + \frac{1}{21} \cdot \frac{1}{2} + \frac{1}{21} \cdot \frac{1}{2} = \frac{1}{2 \cdot 7} \ .$$

The same holds for (R, R), (R, R), and (R, R) (cf. Definition 5). Figure 3 shows the corresponding leaves in a squared pattern and lined pattern, respectively.

If a triple has a different form than considered above, e.g., (B, B), (B, R), and (B, R) then, again by Definition 5, we have for the total amplitude

$$\frac{1}{21} \cdot \frac{1}{2} - \frac{1}{21} \cdot \frac{1}{4} - \frac{1}{21} \cdot \frac{1}{4} = 0 \ .$$

One easily verifies that all remaining total amplitudes are also 0. Finally, we perform the measurement. Clearly, for each leaf which has a total amplitude 0 the measurement results in $\|0\|_p = 0$. For the remaining leaves we obtain $\|\frac{1}{2\cdot 7}\|_p$ which is 1 for every prime p such that $p \notin \{2, 7\}$. If $p = 2$ then we have $\|\frac{1}{2\cdot 7}\|_2 = 2$ and for $p = 7$ we get $\|\frac{1}{2\cdot 7}\|_7 = 7$.

By Lemma 1 there must be at least one line such that all nodes have the same color, and by Lemma 2 it is not possible to have a line colored in RED and a line colored in BLUE simultaneously. So at least one node has p-probability at least 1, and the result output is correct by the definition of the class $\mathcal{U}_7$.

If there are several lines colored in the same color then distinct but correct results may be produced, since any two lines share exactly one point. Thus, the resulting p-probability is always no less than 1. $\square$

The idea of this paper can be extended to obtain even more spectacular advantages of ultrametric algorithms over nondeterministic ones. It is proved that there exist finite projective geometries with n^2+n+1 points and n^2+n+1

lines such that any two lines have exactly one common point and any two points lie on a common line. This allows us to construct a class $\mathcal{U}_m$ of recursive functions similar to the class $\mathcal{U}_7$ above, where $m =_{df} q^2 + q + 1$ for any prime power q. The counterpart of Lemma 1 does not hold but this demands only an additional requirement for the function in the class to have a line colored in one color. Due to the lack of space, we have to omit these results here, but refer the interested reader to [9].

4 Conclusions

In this paper we have studied active learning of classes of recursive functions from value queries. We compared the query complexity of deterministic, nondeterministic, probabilistic, and ultrametric QIM and showed the somehow unexpected result that p-ultrametric QIM can learn classes of recursive function with significantly fewer queries than nondeterministic, probabilistic QIM can do.

References

[1] Ambainis, A., Apsītis, K., Freivalds, R., Smith, C.H.: Hierarchies of probabilistic and team *FIN*-learning. Theoret. Comput. Sci. 261(1), 91–117 (2001)
[2] Ambainis, A., Smotrovs, J.: Enumerable classes of total recursive functions: Complexity of inductive inference. In: Algorithmic Learning Theory, 4th International Workshop on Analogical and Inductive Inference, AII '94, 5th International Workshop on Algorithmic Learning Theory, ALT '94, Reinhardsbrunn Castle, Germany, October 1994, Proceedings. Lecture Notes in Artificial Intelligence, vol. 872, pp. 10–25. Springer-Verlag (1994)
[3] Angluin, D.: Queries and concept learning. Machine Learning 2(4), 319–342 (1988)
[4] Balodis, K., Beriņa, A., Cīpola, K., Dimitrijevs, M., Iraids, J., Jēriņs, K., Kacs, V., Kalējs, J., Krilšauks, R., Lukstiņš, K., Raumanis, R., Scegulnaja, I., Somova, N., Vanaga, A., Freivalds, R.: On the state complexity of ultrametric finite automata. unpublished manuscript (2013)
[5] Denis, F., Lemay, A., Terlutte, A.: Learning regular languages using non deterministic finite automata. In: Grammatical Inference: Algorithms and Applications, 5th International Colloquium, ICGI 2000, Lisbon, Portugal, September 11-13, 2000, Proceedings. Lecture Notes in Artificial Intelligence, vol. 1891, pp. 39–50. Springer-Verlag (2000)
[6] Dragovich, B., Dragovich, A.Y.: A p-adic model of DNA sequence and genetic code. p-Adic Numbers, Ultrametric Analysis, and Applications 1(1), 34–41 (2009)
[7] Freivalds, R.: Ultrametric automata and Turing machines. In: Voronkov, A. (ed.) Turing-100. EPiC Series, vol. 10, pp. 98–112. EasyChair (2012)
[8] Freivalds, R., Bārzdiņš, J., Podnieks, K.: Inductive inference of recursive functions: Complexity bounds. In: Baltic Computer Science, Lecture Notes in Computer Science, vol. 502, pp. 111–155. Springer, Berlin (1991)
[9] Freivalds, R., Zeugmann, T.: Active learning of classes of recursive functions by ultrametric algorithms. Tech. Rep. TCS-TR-A-13-68, Division of Computer Science, Hokkaido University (2013)

[10] Freivalds, R., Kinber, E., Smith, C.H.: On the impact of forgetting on learning machines. J. ACM 42(6), 1146–1168 (1995)
[11] García, P., de Parga, M.V., Álvarez, G.I., Ruiz, J.: Learning regular languages using nondeterministic finite automata. In: Implementation and Applications of Automata, 13th International Conference, CIAA 2008, San Francisco, California, USA, July 21-24, 2008. Proceedings. Lecture Notes in Computer Science, vol. 5148, pp. 92–101. Springer (2008)
[12] Gold, E.M.: Language identification in the limit. Inform. Control 10(5), 447–474 (1967)
[13] Gouvea, F.Q.: p-adic Numbers: An Introduction (Universitext). Springer, Berlin, second edn. (1983)
[14] Greitāne, I.: Probabilistic inductive inference of indices in enumerable classes of total recursive functions. In: Analogical and Inductive Inference, International Workshop AII '89, Reinhardsbrunn Castle, GDR, October 1989, Proceedings. Lecture Notes in Artificial Intelligence, vol. 397, pp. 277–287. Springer (1989)
[15] Jēriņš, K., Balodis, K., Krišlauks, R., Cīpola, K., Freivalds, R.: Ultrametric query algorithms. unpublished manuscript (2012)
[16] Khrennikov, A.: Non-Archimedean Analysis: Quantum Paradoxes, Dynamical Systems and Biological Models. Kluwer Academic Publishers (1997)
[17] Kinber, E., Zeugmann, T.: One-sided error probabilistic inductive inference and reliable frequency identification. Inform. Comput. 92(2), 253–284 (1991)
[18] Koblitz, N.: p-adic Numbers, p-adic Analysis, and Zeta-Functions. Springer, Berlin, second edn. (1984)
[19] Kozyrev, S.V.: Ultrametric analysis and interbasin kinetics. In: 2nd International Conference on p-Adic Mathematical Physics, Belgrade, Serbia and Montenegro, 15 - 21 September 2005. American Institute of Physics Conference Proceedings, vol. 826, pp. 121–128 (2006)
[20] Meserve, B.E.: Fundamental Concepts of Geometry. Dover Publications, New York (1983)
[21] Meyer, L.: Probabilistic language learning under monotonicity constraint. Theoret. Comput. Sci. 185(1), 81–128 (1997)
[22] Meyer, L.: Aspects of complexity of probabilistic learning under monotonicity constraints. Theoret. Comput. Sci. 268(2), 275–322 (2001)
[23] Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information. Cambridge University Press (2000)
[24] Ostrowski, A.: Über einige Lösungen der Funktionalgleichung $\varphi(x)\cdot\varphi(y) = \varphi(xy)$. Acta Mathematica 41(1), 271–284 (1916)
[25] Pitt, L.: Probabilistic inductive inference. J. ACM 36(2), 383–433 (1989)
[26] Pitt, L., Smith, C.H.: Probability and plurality for aggregations of learning machines. Inform. Comput. 77(1), 77–92 (1988)
[27] Rogers, Jr., H.: Theory of Recursive Functions and Effective Computability. McGraw-Hill (1967), reprinted, MIT Press 1987.
[28] Vladimirov, V.S., Volovich, I.V., Zelenov, E.I.: P-Adic Analysis and Mathematical Physics. World Scientific, Singapore (1994)
[29] Wu, Y.C., Lee, Y.S., Yang, J.C., Yen, S.J.: An integrated deterministic and nondeterministic inference algorithm for sequential labeling. In: Information Retrieval Technology 6th Asia Information Retrieval Societies Conference, AIRS 2010, Taipei, Taiwan, December 1-3, 2010. Proceedings. Lecture Notes in Computer Science, vol. 6458, pp. 221–230. Springer (2010)
[30] Zeugmann, T., Zilles, S.: Learning recursive functions: A survey. Theoret. Comput. Sci. 397(1-3), 4–56 (2008)